UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/762,000	01/21/2004	Scott J. Broussard	AUS920030872US1	4404

65362 7590 11/25/2009

HAMILTON & TERRILE, LLP
IBM Austin
P.O. BOX 203518
AUSTIN, TX 78720

EXAMINER	
TO, JENNIFER N	

ART UNIT	PAPER NUMBER
2195	

NOTIFICATION DATE	DELIVERY MODE
11/25/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

tmunoz@hamiltonterrile.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/762,000
Filing Date: January 21, 2004
Appellant(s): BROUSSARD ET AL.

For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 09/10/2009 appealing from the Office action mailed on 1/12/2009.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

WITHDRAWN REJECTIONS

The 35 U.S.C. 101 ground of rejection is not presented for review on appeal because the 35 U.S.C. 101 rejection for Claims 9 and 13-16 have been withdrawn by the examiner.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Patent No. 7203944 VAN RIETSCHOTE ET AL 07-2003

ZHU ET AL. "JESSICA2: A DISTRIBUTED JAVA VIRTUAL MACHINE WITH TRANSPARENT THREAD MIGRATION SUPPORT", IEEE, 2002, PAGES 381-388.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1, 5-9, 13-17, and 21-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over van Rietschote et al. (hereafter Rietschote) (U.S. Patent No. 7203944), in view of Zhu et al. (hereafter Zhu) (Jessica2: A Distributed Java Virtual Machine with transparent Thread Migration Support", IEEE, 2002, pages 381-388).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 5-9, 13-17, and 21-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over van Rietschote et al. (hereafter Rietschote) (U.S. Patent No. 7203944), as applied in claims 1, 3 above, and in view of Zhu et al. (hereafter Zhu) (Jessica2: A Distributed Java Virtual Machine with transparent Thread Migration Support", IEEE, 2002, pages 381-388).

Rietschote and Zhu were cited in IDS filed 05/30/2008.

As per claim 1, Rietschote teaches the invention substantially as claim including a method for operating a virtual machine within a data processing system (abstract), the method comprising the computer-implemented steps of:

running a plurality of virtual machines on one or more devices within the data processing system, wherein each virtual machine in the plurality of virtual machines incorporates functionality for interoperating with other virtual machine in a virtual machine cluster (abstract; col. 2, lines 11-15; col. 3, lines 1-5);

associating the plurality of virtual machines in a virtual machine cluster, wherein each virtual machine in the virtual machine cluster acts as a node within the virtual machine cluster (abstract; col. 2, lines 15-16; col. 3, line 6 through col. 4, line 47);

sharing information about the plurality of virtual machines within the virtual machine cluster such that a virtual machine may be added to the virtual machine cluster or such that a virtual machine may be removed from the virtual machine cluster as the plurality of virtual machines continues to run (abstract; col. 2, lines 16-24; col. 4, line 59 through col. 5, line 21; col. 6, lines 64-66; col. 7, lines 64-66), sharing information further comprising sharing load values representing computer resource utilization among the virtual machines in the virtual machine cluster (col. 4, line 66 through col. 6, line 55); performing a load balancing operating across the virtual machine cluster (col. 4, line 66 through col. 6, line 55); and

determining that a CPU load utilization on a first virtual machine exceeds a threshold value (col. 4, line 66 through col. 6, line 55).

Rietschote did not specifically teach moving a thread from the first virtual machine to a second virtual machine during a load balancing operating.

However, Zhu teaches moving a thread from the first virtual machine to a second virtual machine during a load balancing operating (fig. 1; section 2, i.e. the java threads in the application can migrate from one node to another upon receiving requests from the load monitor).

It would have been obvious to one of an ordinary skill in the art at the time the invention was made to have combined the teaching of Rietschote and Zhu because Rietschote teaches a system for migrating virtual machines among computer systems to balance the load, and Zhu also teaching a system for support the migration of Java Virtual Machine. In addition Zhu teaching of moving a thread from the first virtual machine to a second virtual machine during a load balancing operating would improve the integrity of Rietschote by providing a true parallel execution environment for multithreaded Java application in a distributed Java Virtual Machine (Zhu, abstract).

As per claim 5, Rietschote teaches determining that a memory load utilization on the first virtual machine exceeds a threshold value (col. 10, lines 13-57). Rietschote did not specifically teach moving a set of one or more objects from the first virtual machine to a second virtual machine during a load-balancing operation.

However, Zhu teaches moving a set of one or more objects from the first virtual machine to a second virtual machine during a load-balancing operation (section 3, migrate all the objects used by the migrated thread to another node; section 4.1).

It would have been obvious to one of an ordinary skill in the art at the time the invention was made to have combined the teaching of Rietschote and Zhu because Rietschote teaches a system for migrating virtual machines among computer systems to balance the load, and Zhu also teaching a system for support the migration of Java Virtual Machine. In addition Zhu teaching of moving a set of one or more objects from the first virtual machine to a second virtual machine during a load-balancing operation would improve the integrity of Rietschote by providing a true parallel execution environment for multithreaded Java application in a distributed Java Virtual Machine (Zhu, abstract).

As per claims 6-7, they are rejected for the same reason as claim 1 above.

As per claim 8, Rietschote teaches the invention substantially as claimed in claim 1. Rietschote did not specifically teach running a multithread application within the virtual machine cluster, and dispatching threads of the multithreaded application on different virtual machines such that execution of the multithreaded application spans multiple virtual machines.

However, Zhu teaches running a multithread application within the virtual machine cluster, and dispatching threads of the multithreaded application on different virtual machines such that execution of the multithreaded application spans multiple virtual machines (abstract; figs 1-2; sections 2-3).

It would have been obvious to one of an ordinary skill in the art at the time the invention was made to have combined the teaching of Rietschote and Zhu because Rietschote teaches a system for migrating virtual machines among computer systems to balance the load, and Zhu also teaching a system for support the migration of Java Virtual Machine. In addition Zhu teaching of running a multithread application within the virtual machine cluster, and dispatching threads of the multithreaded application on different virtual machines such that execution of the multithreaded application spans multiple virtual machines would improve the integrity of Rietschote by providing a true parallel execution environment for multithreaded Java application in a distributed Java Virtual Machine (Zhu, abstract).

As per claims 9, 13-17, and 21-24, they are rejected for the same reason as claims 1, 5-8 above.

(10) Response to Argument

(a) In the remarks, Appellant argues that Rietschote fails to teach sharing information about a plurality of virtual machines within the virtual machine cluster such that a virtual machine may be added to the virtual machine cluster or such that a virtual

machine may be removed from the virtual machine cluster as the plurality of virtual machines continues to run.

As to argument (a), examiner respectful disagrees with the Appellant. First, Rietschote disclosed the system have more than one cluster of virtual machines, each group is being managed by the virtual machine kernel(col. 3, lines 1-5, cluster of virtual machine 16A and 16B, cluster of virtual machine 16C and 16 D, the last cluster only one virtual machine 16E), each of the cluster of virtual machine is belong to a computer system (fig. 1, cluster of VM 16 A and VM 16B belong to computer system 10A, cluster of VM 16C and VM 16D belong to computer system 10B, and so forth), and the load on the computer system (for example, computer system 10B of fig. 1) is calculated by the virtual machine kernel 18B or other software within the virtual machine kernel based on the sum of the loads of the virtual machines (VM 16C and VM 16D) (col. 5, lines 14-16; col. 8, lines 24-27, 34-37). It would have been obvious to one of the ordinary skill in the art at the time the invention was made to have recognized that in order for the virtual machine kernel 18B or the other software within the virtual machine kernel to calculate the computer system load of the virtual machines 16C and 16D, the virtual machines 16C and 16D must reported/shared its load with the virtual machine kernel 18B or the other software within the virtual machine kernel. Thus, by reporting the load with the virtual machine kernel 18B or the other software within the virtual machine kernel, the virtual machines 16C and 16D are indirectly sharing their load information through the virtual machine kernel 18B or the other software within the virtual machine kernel. In addition, the claimed language did not clear as to the sharing information between the

virtual machine within the virtual machine cluster is directly (virtual machine directly to virtual machine) or indirectly (for example through virtual machine kernel or other software within the virtual machine kernel as disclosed by Rietchoe). Thus, Rietschote teaches sharing information about a plurality of virtual machines within the virtual machine cluster. Second, Rietschote also teaches in col. 8, lines 9-10 that the migrating process will be activated when the computer system exceed its desire threshold (exceed the load) which cluster result in a virtual machine may be added to the virtual machine cluster or such that a virtual machine may be removed from the virtual machine cluster as the plurality of virtual machines continues to run (abstract; col. 2, lines 16-24; col. 4, line 59 through col. 5, line 21; col. 6, lines 64-66; col. 7, lines 64-66). Therefore, based on the first and second reason explanation above, Rietschote teaches sharing information about a plurality of virtual machines within the virtual machine cluster such that a virtual machine may be added to the virtual machine cluster or such that a virtual machine may be removed from the virtual machine cluster as the plurality of virtual machines continues to run.

(b) In the remarks, Appellant argues that Rietschote and Zhu fail to teach determining that a CPU load utilization on the first virtual machine exceeds a threshold value and moving a thread from the first virtual machine to a second virtual machine during a load balancing operation in response to the first virtual machine exceeding the threshold value.

As to argument (b), examiner respectful disagrees with the Appellant. First Rietschoe disclosed on col. 8, lines 35-37, that the load of each virtual machine is

calculated. Second, Rietschote teaches determining that a CPU load utilization on the first virtual machine exceeds a threshold value (fig. 1; col. 4, line 66 through col. 5, lines 20, 38-42, based on fig. 1, the computer system 10N only have the virtual machine 16E, the load of the computer system 10N in this case is the load of virtual machine, col. 8, lines 10-38, if the computer system load exceed the desired load (exceeds a threshold value), then the migrating process is activated). Thus based on the first and second , Rietschoe teaches determining that a CPU load utilization on the first virtual machine exceeds a threshold value and performing migrating process to balance the workload when the computer system (or the virtual machine in the case of computer system 10N only have one virtual machine 16E) exceed the threshold value (i.e. overload). Rietschoe did not specifically teaches the migrating process performing by moving a thread from the first virtual machine to a second virtual machine during a load balancing operation in response to the first virtual machine exceeding the threshold value. Zhu teaches that based on the load condition (it is well known in the art that the load condition including overload/under load) the java threads in the virtual machine can migrate from one virtual machine to another (fig. 1; section 2). It would have been obvious to one of an ordinary skill in the art at the time the invention was made to have combined the teaching of Rietschote and Zhu because Rietschote teaches a system for activating a migrating process to balance the load when the load exceed/under a desired load level (col. 4, lines 59-65; col. 8, lines 5-12), Zhu teaching a system for support the migration process when the load condition occur (i.e. overload/under load), and Zhu details in the performing of migrating process by moving a thread from the first

virtual machine to a second virtual machine during a load balancing operating would improve the integrity of Rietschote by providing a true parallel execution environment for multithreaded Java application in a distributed Java Virtual Machine (Zhu, abstract), and expanding the utility of Rietschote's system into thread level. Thus the combination of Rietschote and Zhu teaches determining that a CPU load utilization on the first virtual machine exceeds a threshold value and moving a thread from the first virtual machine to a second virtual machine during a load balancing operation in response to the first virtual machine exceeding the threshold value.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Jennifer To/

Patent Examiner, AU 2195

Conferees:

/Meng-Ai An/

Supervisory Patent Examiner, Art Unit 2195

/Eddie C. Lee/

Quality Assurance Specialist, TC 2100